

```

pragma solidity ^0.5.3;

contract Global{

    //1 - blockhash
    function getBlkHash(uint _blockNumber) public view
returns(bytes32)
    {
        return blockhash(_blockNumber);
    }

    //2 - block.coinbase

    function getMinAddr() public view returns(address payable){
        return block.coinbase;
    }

    //3 - block.difficulty

    function getBlkGasLimit() public view returns(uint){
        return block.gaslimit;
    }

    //4 - block._blockNumber

    function getBlkNum() public view returns(uint){
        return block.number;
    }

    // 5- block.timestamp;

    function getBlkTS() public view returns(uint){
        return block.timestamp;
    }

    //6 - gasleft
    function gasLeft() public view returns(uint256){
        return gasleft();
    }

    //7 - msg.data

    function getMsgData(bytes memory _var1) public pure returns(bytes
memory){
        _var1 = 'a';
        return msg.data;
    }

    //8 - msg.sender
    function getMsgSender() public view returns(address){

```

```

        return msg.sender;
    }

    //9 - msg.sig
    function getMsgSig() public pure returns(bytes4){
        return msg.sig;
    }

    //10 - msg.value
    function setMsgValue() public payable returns(uint){
        return msg.value;
    }

    //11 - now
    function getBlkTSNow() public view returns(uint){
        return now;
    }
    //12 - tx.gasprice

    function getGasPrice() public view returns(uint){
        return tx.gasprice;
    }

    //13 - tx.origin

    function getOriginAddr() public view returns(address){
        return tx.origin;
    }

    // Mathematical & Cryptographic functions

    //14 - addmod = (3+5) % 6 = 8 % 6 = 2
    function getAddMod(uint x, uint y, uint k) public pure
returns(uint){
        return addmod(x,y,k);
    }

    //15 - mulmod = (3*5) % 6 = 15 % 6 = 3
    //14 - addmod
    function getMulMod(uint x, uint y, uint k) public pure
returns(uint){
        return mulmod(x,y,k);
    }

    //16 keccak256

    function getKaccak256(bytes memory _input) public pure
returns(bytes32 _output){
        return keccak256(_input);
    }

```

```

//Contract related

//17 - this
function getThis() public view returns(uint){
    return address(this).balance;
}

//18 - selfdestruct
function setDestructContract(address payable _address) public{
    selfdestruct(_address);
}

//ABI functions

//19 - abi.encode

    //- ABI functions

//19 - abi.encode

function getAbiEncode() public pure returns(bytes memory){
    return abi.encode("abc", "def");
}

//20 - abi.encodePacked

function getAbiEncodePacked() public pure returns(bytes memory){
    return abi.encodePacked("abc", "def");
}

function getKeccak256AEP() public pure returns(uint){
    return uint(keccak256(abi.encodePacked("abc", "def")));
}

}

```